Machine Learning (Theory)

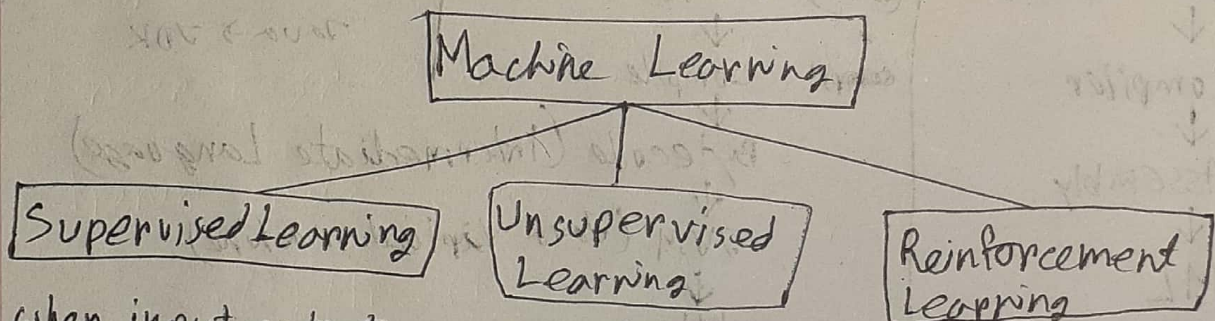
15/1/23

Cls-1

→ কাল lab হবে না,

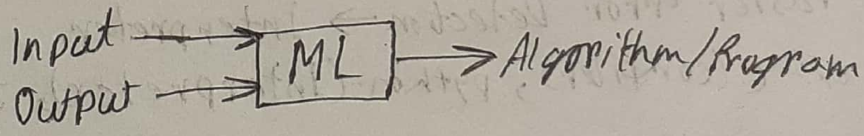
Machine Learning :-

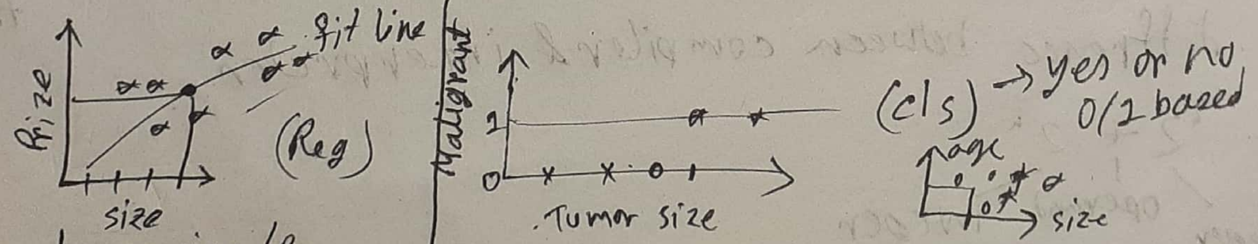
* A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience. — Tom Michall (1998)
* Arthur Samuel (1959) :-



- **Supervised Learning:** when input, output given & feature leveled.
- **Unsupervised Learning:** No output, only input & feature not leveled.
- **Reinforcement Learning:** when machine learns through trial & errors.

* **Supervised learning:-**





house size / Prize

finding

Supervised learning gets :-
- → Regression (Continous) - value ~~finding~~
- → Classification (Discreate) - given class / identify করা

ML (cls-2) T-2

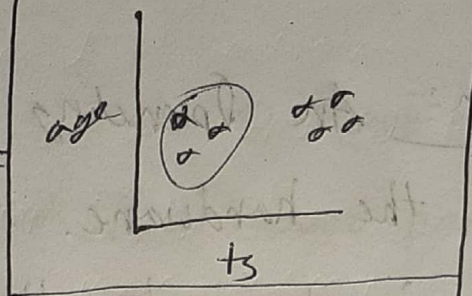
18/1/23

#Supervised Learning :-

Input (x)	Output (y)
1	5
3	10
5	
10	

Modal / H/F

Unsupervised :-



#Reinforcement :-

Action so against reward point 2nnv

Regression :- output space continuous

classification :- " " discreate

* Regression :-

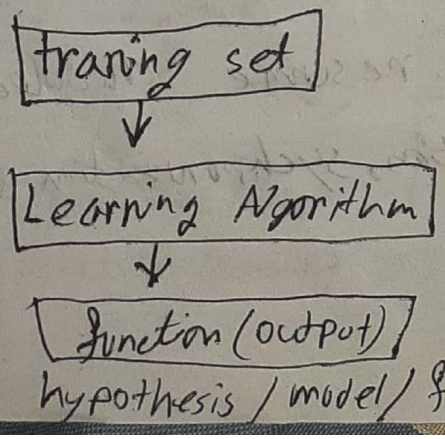
Linear Regression → stright line draw ase

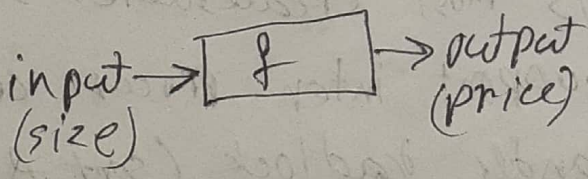
* Traning Data :- (House size & Price)

x size (ft^2)	Price (K) y
2000	400
1500	350
1200	280
500	150
⋮	⋮

$n = 50$

x = input
y = output
n = number of data
(x^i, y^i) = i^{th} traning example
$(x^3, y^3) = (1200, 280)$
x = input $\hat{y}$ = predition

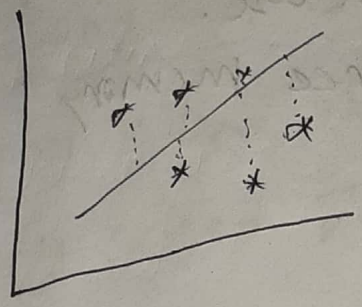


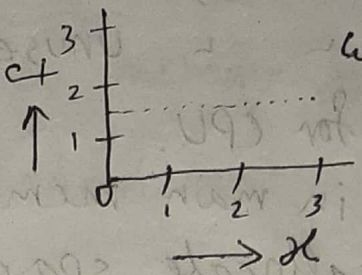


Linear Regression:

$f = wx + b$ [w = slope, b = constant]

w, b : parameters, so they are changeable

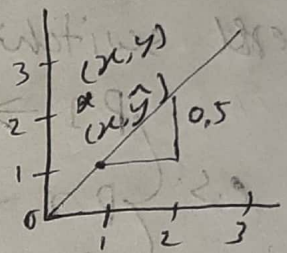




$w = 0, b = 1.5$

Cost function:

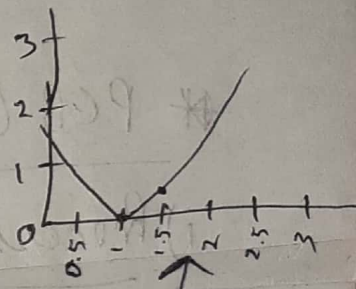
$f_{w,b} = wx + b$



Squared error cost function:-

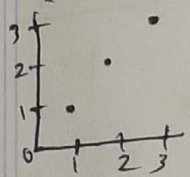
$$J_{w,b} = \frac{1}{2n} \sum_{i=1}^{n} (\hat{y}^{(i)} - y^{(i)})^2$$

$$J(w,b) = \frac{1}{2n} \sum_{i=1}^{n} (f_{w,b}(x^{(i)}) - y^{(i)})^2$$



Objective:- to minimize cost function (J).

Task:- $f_w = wx$, where $b = 0$; $J_w = \frac{1}{2n} \sum_{i=1}^{n} (f(x^{(i)}) - (y^{(i)}))^2$



$w = 1$:- $J_1 = \frac{1}{2*3}(1-1)^2 + (2-2)^2 + (3-3)^2 = 0$

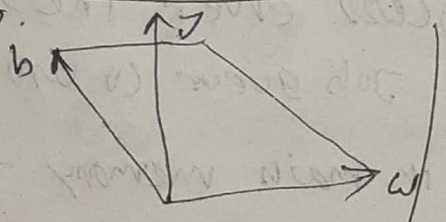
$w = 0.5$:- $J_{0.5} = \frac{1}{2*3}(0.5-1)^2 + (1-2)^2 + (1.5-3)^2 = 0.58$

$w = 1.5$:- $J_{1.5} = \frac{1}{2*3}(1.5-1)^2 + (3-2)^2 + (4.5-3)^2 = 0.58$

$w = 0$:- $J_0 = \frac{1}{2*3}(0-1)^2 + (0-2)^2 + (0-3)^2 = 2.33$

Gradient Descent Algorithm:- used for cost / loss function optimization.

↳ Iterative step.



$$w_{new} = w_{old} - \alpha \frac{d}{dw} J(w,b)$$

↓ learning rate

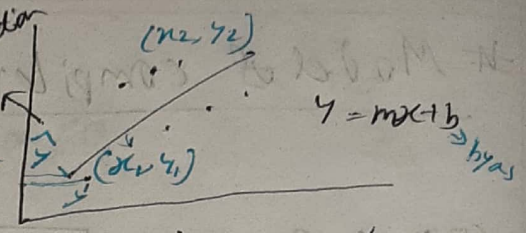
$$b_{new} = b_{old} - \alpha \frac{d}{db} J(w,b)$$

ML (T-3)

23/1/23

Linear Regression :-

→ Squared Error :- call করে best fit line বের করা



$$SE = (y_1 - (mx_1 + b))^2 + (y_2 - (mx_2 + b))^2 + \dots + (y_n - (mx_n + b))^2$$

(y_1 → true value; SE → minimization করতে হবে)

$$= y_1^2 + 2y_1(mx_1 + b) + (mx_1 + b)^2 + y_2^2 + 2y_2(mx_2 + b) + (mx_2 + b)^2 + \dots + y_n^2 - 2y_n(mx_2 + b) + (mx_n + b)^2$$

(m, b বের করতে হবে)

Simplify করি

$$SE = (y_1^2 + y_2^2 + \dots + y_n)^2 - 2m(x_1y_1 + x_2y_2 + \dots + x_ny_n) - 2b(y_1 + y_2 + \dots + y_n) + m^2(x_1^2 + x_2^2 + \dots + x_n^2) + 2mb(x_1 + x_2 + \dots + x_n) - nb^2$$

Here, $\frac{y_1^2 + y_2^2 + \dots + y_n^2}{n} = \overline{y^2}$ (average of y^2)

$$y_1^2 + y_2^2 + \dots + y_n^2 = n\overline{y^2} \quad \text{---- (1)}$$

again,

$$x_1y_1 + x_2y_2 + \dots + x_ny_n = n\bar{x}\bar{y} \quad \text{---- (2)}$$

$$SE = n\overline{y^2} - 2mn\bar{x}\bar{y} - 2bn\bar{y} + m^2n\overline{x^2} + 2mbn\bar{x} + nb^2$$

At minimum SE :- $\frac{SE}{\partial m} = 0$

$$\Rightarrow 0 - 2n\bar{x}\bar{y} + 2nx^2m + 2bn\bar{x} = 0$$

[2n দ্বারা divide] $\Rightarrow m\overline{x^2} + b\bar{x} = \overline{xy}$ ---- (3)

$$\frac{\delta SSE}{\delta m} = 0$$

$$\Rightarrow -2n\bar{y} + 2n\bar{x^2}m + 2bn\bar{x} = 0$$

$$\Rightarrow m\bar{x} + b = \bar{y} \quad (2)$$

$$\Rightarrow m = \frac{\bar{x}\bar{y} - \bar{xy}}{(\bar{x})^2 - \bar{x^2}} = \frac{\bar{xy} - \bar{x}\bar{y}}{\bar{x^2} - (\bar{x})^2}$$

$$b = \bar{y} - m\bar{x}$$

Gradient decent Algorithm: (feature-for 1 data-(x))

$$f(x) = y = mx + b = \hat{w}x + b \mid h(x) = \theta_0 + \theta_1 x$$ (weight)

$$= \theta_0 + \theta_1 x_1^{(1)} + \theta_2 x_2^{(2)}$$ ((1) → data number, (2) → feature number)

$$h(\theta) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_d x_d$$ (x_1 → independent)

$$= \sum_{j=1}^{d} \theta_j x_j = \theta^T x \quad [\theta_0, \theta_1, \theta_2 \dots \theta_d] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

Cost function:- $J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

(prediction: $h_\theta(x^{(i)})$, true: $y^{(i)}$)

objective:- θ^* = argument minimum $J(\theta)$

GDA

$$\theta_j^{new} = \theta_j^{old} - \alpha \frac{\delta J(\theta)}{\delta \theta_j}$$

α → learning rate; $\frac{\delta J(\theta)}{\delta \theta_j}$ → rate of change/slope

Batch Gradient Descent:

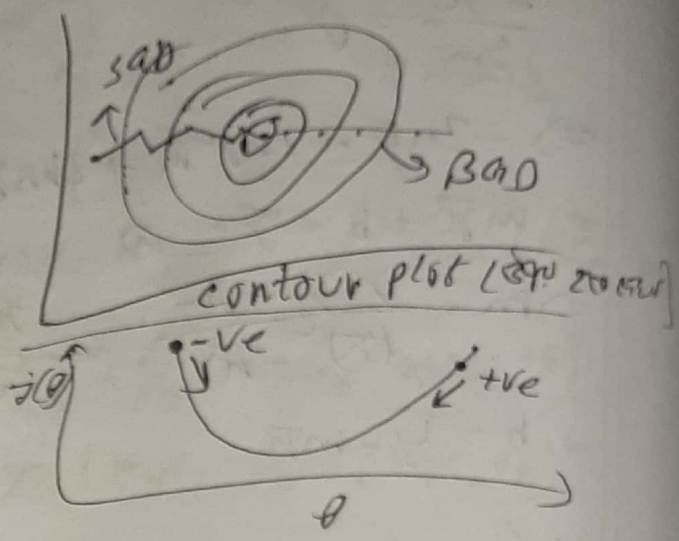
Full data at a time

~~# Stochastic~~

Stochastic Gradient Descent:

At a time one by one update.

Mini batch: packet wise



* For single Point:

$$n=1 \quad \frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x)-y)^2$$

$$= \frac{1}{2} \cdot 2 (h_\theta(x)-y) \frac{\partial}{\partial \theta} (h_\theta(x)-y)$$

$$= (h_\theta(x)-y) x_j$$

$h_\theta(x) = (\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_j x_j + \dots + \theta_d x_d)$

$$\therefore \theta^{old} - \alpha (h_\theta(x)-y) x_j$$

$$\frac{\partial}{\partial \theta_j} \sum_{i=1}^{d} \theta_j x_j - y$$

$$= \frac{\partial}{\partial \theta_j} (\theta_0 x_0 + \theta_1 x_1 + \dots + \theta_j x_j + \dots + \theta_d x_d - y)$$

$$\frac{\partial \theta_j x_j}{\partial \theta_j} = x_j$$

* Repeat until convergence: → GDA

for $i = 1$ to n

$$\theta_j = \theta_j - \alpha (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

↓ GDA.

x^i, y^i = input data

$x_j^{(i)}$ → which nth feature

ML (T-4)

1/2/23

* $h_\theta(x) = \sum_{j=0}^{d} \theta_j x_j = \underbrace{\theta^T}_{s} \underbrace{x}_{v}$ → vector

[d-dr vector = d-dr feature]

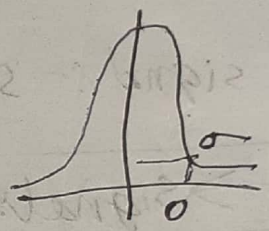
loss function: $J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2$

$x^{(i)} \in \mathbb{R}^{d+1}$
$y^{(i)} \in \mathbb{R}$

Probabilistic Interpretation:

$y^{(i)} = \theta^T x^{(i)} + \underbrace{\epsilon^{(i)}}_{\text{error / noise}}$

$\epsilon^{(i)} \sim N(0, \sigma^2)$
(0 → mean, σ^2 → variance)



$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$ → Gaussian distribution

$p(y^{(i)} | x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$

($x^{(i)}$ → given, θ → fixed)

Maximum Likelihood Estimation:

Likelihood function:

$L(\theta) = p(\vec{y} | x; \theta) = \prod_{i=1}^{n} p(y^{(i)} | x^{(i)}; \theta)$

$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$

~~log~~

log Likelihood :-

$$\ell(\theta) = \log L(\theta)$$

$$= \log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$= n \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \frac{1}{2} \sum_{i=1}^{n} (y^{(i)} - \theta^T x^{(i)})^2$$

→ likelihood ক maximum করলে probability বাড়ে

Here, maximizing $\ell(\theta)$ gives the same answer as minimizing $\frac{1}{2} \sum_{i=1}^{n} (y^{(i)} - \theta^T x^{(i)})^2 = J(\theta)$

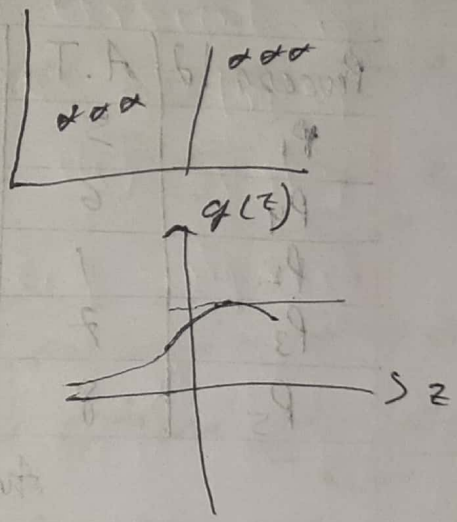
Classification :-

$y = \{0, 1\}$ binary classification

$h_\theta(x) \in [0, 1]$

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$g(z) = \frac{1}{1 + e^{-z}} \quad \text{(sigmoid function)}$$



$$\begin{cases} P(y=1 \mid x; \theta) = h_\theta(x) \\ P(y=0 \mid x; \theta) = 1 - h_\theta(x) \end{cases}$$

$$\rightarrow P(y \mid x; \theta) = h_\theta(x)^y (1 - h_\theta(x))^{1-y}$$

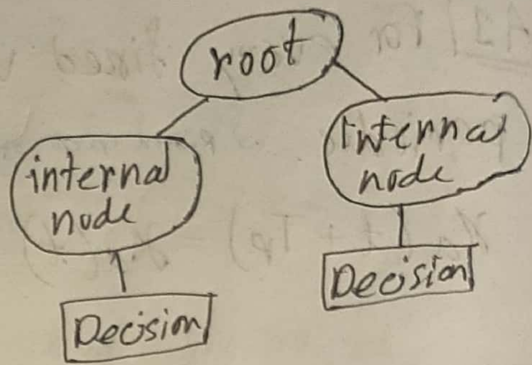
6/2/23

ML (2-3)

Decision Tree:

Q. Tennis খেলা হবে কিনা?

Day	Outlook	Temp	Humidity	Tennis
D_1	Sunny	Hot	High	No
D_2	Sunny	Mild	Normal	Yes
D_3	Rain	Cold	Normal	No



* Information Gain (IG):- [illegible]

$$IG = Entropy(s) - \sum_{i=1}^{c} \frac{|s_i|}{|s|} Entropy(s_i)$$

* Entropy:- [illegible]

It is a measure from information theory, ~~characteristics~~ characterize the (im)purity/homogeneity ~~homogeneity~~ of ~~a~~ arbitrary collection.

$$E = -\sum_{i=1}^{c} p_i * \log_2 (p_i)$$

* Factors affecting sunburn:-

Name	Hair	Hight	Weight	Lotion	Decision
Sara	Blonde	avg	light	No	✓
Dana	Blonde	tall	avg	Yes	✗
Alex	Brown	short	avg	No	✗
Anny	Blonde	short	avg	No	✓
Emily	Red	avg	heavy	No	✓
Pete	Brown	tall	heavy	No	✗
Jhon	Brown	avg	heavy	No	✗
Kati	Blonde	short	light	Yes	✗

~~0.5 - 0.2175~~

$0.5 - 0.2175$

$\frac{S_i}{S} \times \{-\sum P_i \log_2(P_i)\}$

→ avg entropy for hair color $= \frac{4}{8} \times \{-\frac{2}{4}\log_2(\frac{2}{4}) - \frac{2}{4}\log_2(\frac{2}{4})\}$ (Blond)

$+ \frac{3}{8} \times \{-\frac{3}{3}\log_2(\frac{3}{3}) - 0\} + \frac{1}{8} \times \{-\frac{1}{1}\log_2(\frac{1}{1}) - 0\}$ (Brown, Red)

$= 0.5$

→ avg entropy for height $= 2 \times \left[\frac{3}{8} \times \{-\frac{2}{3}\log_2(\frac{2}{3}) - \frac{1}{3}\log_2(\frac{1}{3})\}\right] +$ (avg, short)

$\frac{2}{8} \times \{-\frac{2}{2}\log_2(\frac{2}{2}) - 0\}$ (tall)

$= 0.69$

→ avg entropy for weight $= \frac{2}{8}\{-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})\} +$ (light)

$2 \times \left[\frac{3}{8}\{-\frac{2}{3}\log_2(\frac{2}{3}) - \frac{1}{3}\log_2(\frac{1}{3})\}\right]$ (avg, heavy)

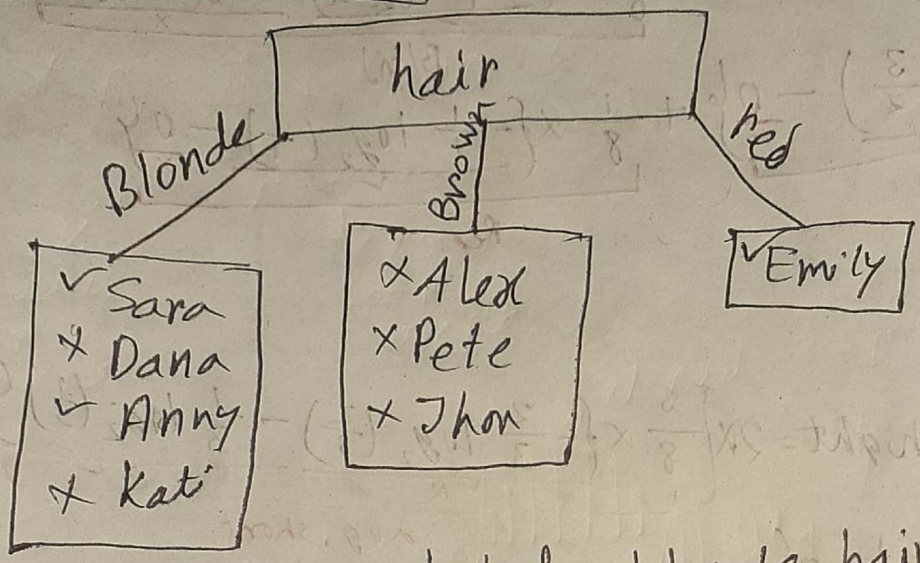
$= 0.936$

→ avg entropy for lotion $= \frac{2}{8}\{-\frac{2}{2}\log_2(\frac{2}{2})\} + \frac{6}{8}\{-\frac{3}{6}\log_2(\frac{3}{6}) - \frac{3}{6}\log_2(\frac{3}{6})\}$ (Yes, No)

$= 0.75$

→ Decision (ES) $= \frac{3}{8}\log_2(\frac{3}{8}) - \frac{5}{8}\log_2(\frac{5}{8})$.

Information Gain :- $E(S) - 0.5$



Another test needed for blonde hair color :-

$$\text{Height} = \frac{2}{4}\left\{-\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right)\right\} + 0 + 0$$

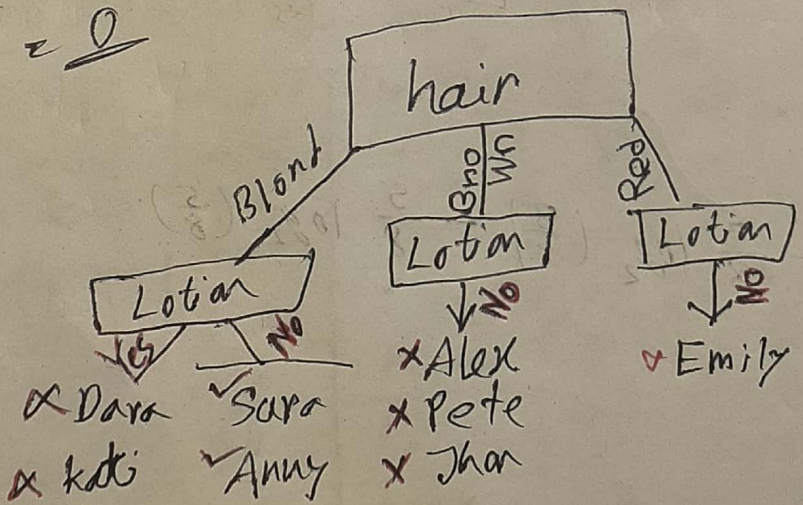
$$= 0.5$$

$$\text{Weight} = 2 \times \left[\frac{2}{4}\left\{-\frac{1}{2}\log\left(\frac{1}{2}\right)\right\} + 0 + 0\right]$$

$$= 1$$

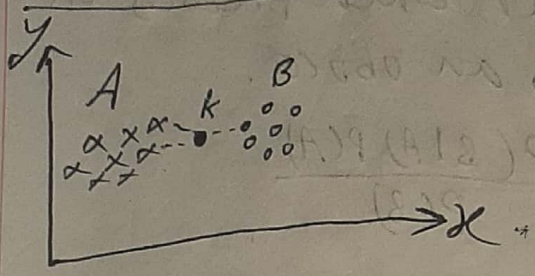
$$\text{Lotion} = 2 \times \left[\frac{2}{4}\left\{-\frac{2}{2}\log\left(\frac{2}{2}\right) - -\log\left(-\right)\right\}\right.$$

$$= 0$$



ML-(T-5) 8/2/23

K-Nearest Neighbors Algorithm (KNN):-



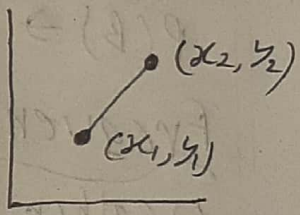
→ k select করতে হবে 1st।
→ যেকোন data point এর distance calculate করতে হবে
→ distance sort
→ ■ Nearest data point কোন set এ বেশি করতে হবে (nb of values close to k)

* Parametric Algorithm:- learning এর প্রক্রিয়া এ করে করলে যেন loss minimum.
* Non-Parametric Algorithm:- Ex:- Decision Tree, KNN
* k এর ৩টি tips:- i) odd number, (ii) কম feature, (iii) medium value.

Distance Measure:-

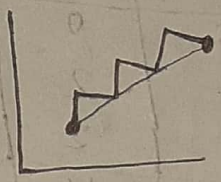
→ Euclidean distance:- দুই point এর দূরত্ব।

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

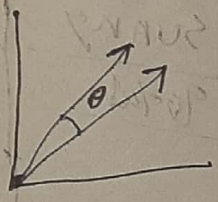


→ Manhatan distance:-

$$|x_1 - x_2| + |y_1 - y_2|$$



→ Cosine similarity $= \frac{|A||B| * \cos\theta}{|A||B|}$ [self study]



Advantage of KNN	Disadvantage of KNN
→ Easy to implement.	→ The algorithm gets significantly slower as the number of examples (training data) and/or predictions (test data) increase.
→ No need to build a model, tune several parameters or make additional assumption.	→ Memory requirement is high.
→ It can be used for classification as well as regression.	

* Application:- Decision Tree, Recomended system, Data processing, classification.

Naive Bayes Algorithm:-

It is a probabilistic classifier, which predicts on the basis of the probability of an object.

Bayes theorem:- $P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$ (A: feature, B: outcome)

where,

P(A|B) → Probability of hypothesis A on the observed event B
Posterior probability

P(B|A) → Probability of ~~hypothesis~~ the evidance given that the probability of hypothesis is true.
likelihood Probability

P(A) → Probability of hypothesis before observing the evidence

P(B) → Probability of evidence.

Frequency table:

Weather	Y (Yes)	N (No)	Probability
Overcast	5	0	5/14 = 0.357
Rainy	2	2	4/14 = 0.28
Sunny	3	2	5/14 = 0.357
Total	10	4	14
	10/14 = 0.71	4/14 = 0.28	

Q. if weather = sunny, then play = Yes / No?

P(S|Y) = 3 / 10 = 0.3 (3 → nb of Y. ins; 10 → Total Y)
P(S) = 5/14 = 0.35
P(Y) = 10/14 = 0.71
P(S|N) = 2/4 = 0.5
P(N) = 4/14 = 0.28

Weather	Play
Rainy	Y
Sunny	Y
Overcast	Y
Overcast	Y
Sunny	N
Rainy	Y
Sunny	Y
Overcast	Y
Rainy	N
Sunny	N
Sunny	Y
Rainy	N
Overcast	Y
Overcast	Y

$$P(Yes|Sunny) = \frac{P(Sunny|Yes) * P(Yes)}{P(Sunny)}$$

$$= \frac{0.3 * 0.71}{0.35} = \underline{0.6}$$

$$P(No|Sunny) = \frac{P(Sunny|No) * P(No)}{P(Sunny)} = \frac{0.5 * 0.28}{0.35}$$

$$= \underline{0.408}$$

since P(Yes|Sunny) > P(No|Sunny) therefore decision = Yes.